Collaborating on Referring Expressions

Peter A. Heeman and Graeme Hirst

Technical Report 435
August 1992

UNIVERSITY OF
ROCHESTER
COMPUTER SCIENCE

Collaborating on Referring Expressions

Peter A. Heeman
Department of Computer Science
University of Rochester*
Rochester, New York
14627
heeman@cs.rochester.edu

Graeme Hirst
Department of Computer Science
University of Toronto
Toronto, Canada
M5S 1A4
gh@cs.toronto.edu

Technical Report 435

August 1992

Abstract

This paper presents a computational model of how conversational participants collaborate in order to make a referring action successful. The model is based on the view of language as goal-directed behavior. We propose that the content of a referring expression can be accounted for by the planning paradigm. Not only does this approach allow the processes of building referring expressions and identifying their referents to be captured by plan construction and plan inference, it also allows us to account for how participants clarify a referring expression by using meta-actions that reason about and manipulate the plan derivation that corresponds to the referring expression. To account for how clarification goals arise and how inferred clarification plans affect the agent, we propose that the agents are in a certain state of mind, and that this state includes an intention to achieve the goal of referring and a plan that the agents are currently considering. It is this mental state that sanctions the adoption of goals and the acceptance of inferred plans, and so acts as a link between understanding and generation.

*This research was carried out while the first author was at the Department of Computer Science, University of Toronto.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

ublic reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, athering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this ollection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson avis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1992	technical report

. TITLE AND SUBTITLE
Collaborating on Referring Expressions

5. FUNDING NUMBERS
ONR/DARPA N00014-92-J-1512

. AUTHOR(S)
Peter A. Heeman and Graeme Hirst

. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Science Dept.
734 Computer Studies Bldg.
University of Rochester
Rochester, NY 14627-0226

8. PERFORMING ORGANIZATION REPORT NUMBER
TR 435 .

. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research	DARPA
Information Systems	1400 Wilson Blvd.
Arlington, VA 22217	Arlington, VA 22209

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

1. SUPPLEMENTARY NOTES

2a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution of this document is unlimited.

12b. DISTRIBUTION CODE

3. ABSTRACT *(Maximum 200 words)*

This paper presents a computational model of how conversational participants collaborate in order to make a referring action successful. The model is based on the view of language as goal-directed behavior. We propose that the content of a referring expression can be accounted for by the planning paradigm. Not only does this approach allow the processes of building referring expressions and identifying their referents to be captured by plan construction and plan inference, it also allows us to account for how participants clarify a referring expression by using meta-actions that reason about and manipulate the plan derivation that corresponds to the referring expression. To account for how clarification goals arise and how inferred clarification plans affect the agent, we propose that the agents are in a certain state of mind, and that this state includes an intention to achieve the goal of referring and a plan that the agents are currently considering. It is this mental state that sanctions the adoption of goals and the acceptance of inferred plans, and so acts as a link between understanding and generation.

14. SUBJECT TERMS
referring expressions; collaboration; planning; discourse

15. NUMBER OF PAGES
33 pages

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	UL

1 Introduction

People are goal oriented and can plan courses of actions to achieve their goals. But sometimes they might lack the knowledge needed to formulate a plan of action, or some of the actions that they plan might depend on coordinating their activity with other agents. How do they cope? One way is to work together, or *collaborate*, in formulating a plan of action with other people who are involved in the actions or who know the relevant information.

Even in the apparently simple linguistic task of referring, in an utterance, to some object or idea can involve exactly this kind of activity: a collaboration between the speaker and the hearer. The speaker has the goal of the hearer identifying the object that the speaker has in mind. The speaker attempts to achieve this goal by constructing a description of the object that she thinks will enable the hearer to identify it. But since the speaker and the hearer will inevitably have different beliefs about the world, the hearer might not be able to identify the object. Often, when the hearer cannot do so, the speaker and hearer collaborate in making a new referring expression that accomplishes the goal.

This paper presents a computational model of how a conversational participant collaborates in making a referring action successful. We use as our basis the model proposed by Clark and Wilkes-Gibbs (1986), which gives a descriptive account of the conversational moves that participants make when collaborating upon a referring expression. We cast their work into a model based on the planning paradigm.

We propose that referring expressions can be represented by plan derivations, and that plan construction and plan inference can be used to generate and understand them. Not only does this approach allow the processes of *building* referring expressions and *identifying* their referents to be captured in the planning paradigm, it also allows us to use the planning paradigm to account for how participants *clarify* a referring expression. In this case, we use meta-actions that encode how a plan derivation corresponding to a referring expression can be reasoned about and manipulated.

To complete the picture, we also need to account for the fact that the conversants are *collaborating*. We propose that the agents are in a mental state that includes not only an intention to achieve the goal of the collaborative activity but also a plan that the participants are currently considering. In the case of referring, this will be the plan derivation that corresponds to the referring expression. This plan is in the common ground of the participants, and we propose rules that are sanctioned by the mental state both for *accepting* plans that clarify the current plan, and for *adopting* goals to do likewise. The acceptance of a clarification results in the current plan being updated. So, it is these rules that specify how plan inference and plan construction affect and are affected by the mental state of the agent. Thus, the mental state, together with the rules, provides the link between these two processes. An important consequence of our proposal is that the current plan need not allow the successful achievement of the goal. Likewise, the clarifications that agents propose need not result in a successful plan in order for them to be accepted.

As can be seen, our approach consists of two tiers. The first tier is the planning component, which accounts for how utterances are both understood and generated. Using the planning paradigm has several advantages: it allows both tasks to be captured in a single paradigm that is used for modeling general intelligent behavior; it allows more of the content of an utterance to be accounted for by a uniform process; and only a single knowledge source for referring expressions is needed instead of having this knowledge embedded in special algorithms for each task. The second tier accounts for the collaborative behavior of the agents: how they adopt goals and coordinate their activity. It provides the link between the mental state of the agent and the planning processes.

In accounting for how agents collaborate in making a referring action, our work aims to make the following contributions to the field. First, although much work has been done on how agents request clarifications, or respond to such requests, little attention has been paid to the collaborative aspects of clarification discourse. Our work attempts a plan-based formalization of what linguistic collaboration is, both in terms of the goals and intentions that underlie it and the surface speech acts that result from it. Second, we address the act of referring and show how it can be better accounted for by the planning paradigm. Third, previous plan-based linguistic research has concentrated on either construction or understanding of utterances, but not both. By doing both, we will give our work generality in the direction of a complete model of the collaborative process. Finally, by using Clark and Wilkes-Gibbs's model as a basis for our work, we aim not only to add support to their model, but gain a much richer understanding of the subject.

In order to address the problem that we have set out, we have limited the scope of our work. First, we look at referring expressions in isolation, rather than as part of a larger speech act. Second, we assume that agents have mutual knowledge of the mechanisms of referring expressions and collaboration. Third, we deal with objects that both the speaker and hearer know of, though they might have different beliefs about what propositions hold for these objects. Fourth, as the input and the output to our system, we use representations of surface speech actions, not natural language strings. Finally, although belief revision is an important part of how agents collaborate, we do not explicitly address this.

2 Referring as a Collaborative Process

Clark and Wilkes-Gibbs (1986) investigated how participants in a conversation collaborate in making a referring action successful. They conducted experiments in which participants had to refer to objects—tangram patterns—that are difficult to describe. They found that typically the participant trying to describe a tangram pattern would present an initial referring expression. The other participant would then pass judgment on it, either *accepting* it, *rejecting* it, or *postponing* his decision. If it was rejected or the decision postponed, then one participant or the other would *refashion* the referring expression. This would take the form of either repairing the expression by correcting speech errors, *expanding* it by adding further qualifications, or *replacing* the original expression with a new expression. The referring expression that results from this is then judged, and the process continues until the referring expression is acceptable enough to the participants for current purposes. This final expression is contributed to the participants' common ground.

Below are two excerpts from Clark and Wilkes-Gibbs's experiments that illustrate the acceptance process.

(2.1) **A:** [1] Um, third one is the guy reading with, holding his book to the left.
B: [2] Okay, kind of standing up?
A: [3] Yeah.
B: [4] Okay.

In this dialogue, person A makes an initial presentation in line 1. Person B postpones his decision in line 2 by voicing a *tentative "okay"*, and then proceeds to refashion the referring expression, the result being "the guy reading, holding his book to the left, kind of standing up." A accepts the new expression in line 3, and B signals his acceptance in line 4.

(2.2) **A:** [1] Okay, and the next one is the person that looks like they're carrying something and it's sticking out to the left. It looks like a hat that's upside down.

B: [2] The guy that's pointing to the left again?

A: [3] Yeah, pointing to the left, that's it! (laughs)

B: [4] Okay.

In the second dialogue, B implicitly rejects A's initial presentation by replacing it with a new referring expression in line 2, "the guy that's pointing to the left again." A then accepts the refashioned referring expression in line 3.

An important question is what happens after a refashioning that fails to create a referring expression that allows for the identification of the referent. Does the other participant find the refashioning *move* unacceptable, or is it the resulting *expression* that is found unacceptable? The ramification of this is that with the former view the refashioning move itself would need to be again refashioned, whereas with the latter view, it is the resulting expression that would be refashioned. It is this latter view that is proposed by Clark and Wilkes-Gibbs to account for the acceptance process. Since each judgment and refashioning pair result in a new referring expression replacing the previous one, the only dependence between subsequent pairs and their predecessor is through the referring expression that the predecessor proposed. This leads to an acceptance process that is iterative rather than recursive, and we claim that the most recently proposed referring expression represents the state of the collaborative process. This state is in the common ground of the participants, and the judgment and refashioning moves serve to update the agents' common ground with respect to the collaborative process.

In later work, Clark and Schaefer (1989) propose that "each part of the acceptance phase is itself a contribution" (p. 269), and the acceptance of these contributions depends on whether the hearer "believes he is understanding well enough for current purposes" (p. 267). Although Clark and Schaefer use the term *contribution* with respect to the discourse, rather than the collaborative effort of referring, their proposal is still relevant here: judgments and refashionings are contributions to the collaborative effort and are subjected to an acceptance process, with the result being that once they are accepted, the state of the collaborative activity is updated. So, what constitutes grounds for accepting a judgment or clarification? From the claim for the iterative structure of the acceptance process, we can see that if one agent finds the current referring expression problematic, the other must accept the judgment. Likewise, if one agent proposes a referring expression, through a refashioning, the other must accept the refashioning.

To sum up: in collaborating upon a referring expression, agents use judgment and refashioning moves to further the collaborative effort. These conversational moves are subject to an acceptance process, resulting in the updating of the common ground of the participants, specifically, the referring expression that represents the state of the collaborative effort.

3 Referring Expressions

3.1 Planning and Referring

By viewing language as action, the planning paradigm can be applied to natural language processing. The actions in this case are *speech acts* (Austin, 1962; Searle, 1969), and include such things as promising, informing, and requesting. Cohen and Perrault (1979) developed a system that uses plan construction to map an agent's goals to speech acts, and Allen and Perrault (1980) use plan inference to understand an agent's plan from its speech acts. By viewing it as action (Searle, 1969), referring can be incorporated into a planning model. Cohen's model (1981) planned requests that the hearer identify a referent, whereas Appelt (1985) planned *concept activations*, a generalization of referring actions.

Although acts of reference have been incorporated into plan-based models, determining the content of referring expressions hasn't been. For instance, in Appelt's model, concept activations can be achieved by the action `describe`, which is a primitive, not further decomposed. Rather, this action has an associated procedure that determines a description that satisfies the preconditions of `describe`. Such special procedures have been the mainstay for accounting for the content of referring expressions, both in constructing and in understanding them, as exemplified by Dale (1989), who chose descriptors on the basis of their discriminatory power, Ehud Reiter (1990), who focused on avoiding misleading conversational implicatures when generating descriptions, and Mellish (1985), who used a constraint satisfaction algorithm to identify referents.

Our work follows the plan-based approach to language generation and understanding. We extend the earlier approaches of Cohen and Appelt by accounting for the content of the description at the planning level. This is done by having surface speech actions for each component of a description, plus a surface speech action that expresses a speaker's intention to refer. A referring action is composed of these primitive actions, and the speaker utters them in her attempt to refer to an object.

The surface speech actions are actions that the plan construction and plan inference processes can reason about. These actions have constraints that express conditions under which they can be used to refer to an object; for instance, that it be mutually believed that the object has a certain property (Clark and Marshall, 1981; Perrault and Cohen, 1981; Nadathur and Joshi, 1983). Also, there are intermediate plans that encode the knowledge of how a description can allow a hearer to identify an object, and these ensure that the referring expression includes sufficient descriptors so that the hearer can identify the referent. The intermediate plans do this by having *mental actions* as steps in their decomposition. These mental actions determine which objects could be believed to be the referent of the referring expression. There is a constraint to ensure that a sufficient number of surface speech actions are added so that the set of candidates associated with the entire referring expression consists of only a single object. This allows the plan constructor to know when enough descriptors have been added. Furthermore, the explicit encoding of the adequacy of referring expressions allows referent identification to fall out of the plan inference process. The mental actions are performed on the candidate sets, and the constraints are evaluated, and so the referent can be determined in a manner analogous to a constraint satisfaction algorithm.

Our approach to treating referring as a plan in which surface speech actions correspond to the components of the description allows us to capture how participants collaborate in building a referring expression. Plan repair techniques can be used to refashion an expression if it is not adequate, and clarifications can refer to the part of the plan derivation that is in question or is being repaired. Thus we can model a collaborative dialogue in terms of the changes that are being made to the plan derivation.

The referring expression plans that we propose are not simply data structures, but are mental objects that agents have beliefs about (Pollack, 1990). The plan derivation expresses beliefs of the speaker: how actions contribute to the achievement of the goal, and what constraints hold that will allow successful identification.[1] So plan construction reasons about the beliefs of the agent in constructing a referring plan; likewise, plan inference, after hypothesizing a plan that is consistent with the observed actions, reasons about the other participant's (believed) beliefs in satisfying the constraints of the plan. If the hearer is able to satisfy the constraints, then he will have understood the plan and be able to identify the referent, since a term corresponding to it would have been instantiated in the inferred plan. Otherwise, he would have a constraint that is unsatisfiable, which he takes as being the error in the plan. (We do not reason about how the error affects the satisfiability of the goal of the plan nor use the error to revise the beliefs of the hearer.)

3.2 Vocabulary and Notation

Before we present the plan schemas for referring expressions, we need to introduce some notation that we use. Table 1 summarizes our basic predicates and actions. (Additional notation will be introduced in section 4.2.)

For reasoning about beliefs, we have taken a syntactic approach, with the addition of several inference rules. (The rules can be applied within an arbitrary nesting of belief operators.) The first rule is that for certain types of propositions, if a participant believes the proposition, then he will believe that the other participant also believes it. The second rule is that if a participant believes a proposition and he believes that the other participant also believes it, then he will believe that it is mutually believed. The third rule is for inferring an *alternating belief* (defined in the table). This rule is that if an agent believes something or believes that the other agent has an alternating belief about it, then he will have an alternating belief it (this recursion is applied to the maximum embedding of beliefs in the model). The first and second rules are intended to capture the community membership inferences of Clark and Marshall (1981), and should be made into default rules (cf. Perrault 1990).

Our terminology for planning follows the general literature.[2] We use the terms *action schema*, *plan derivation*, *plan construction*, and *plan inference*. An action schema consists of a *header*, *where*-clauses, *constraints*, a *decomposition*, and an *effect*; and it encodes the constraints under which an effect can be achieved by performing the steps in the decomposition; the where-clauses are used to instantiate such variables as **Speaker** and **Hearer**. A plan derivation is an instance of an action that has been recursively expanded into primitive actions—its *yield*. Each component in the plan—the action headers, where-clauses, constraints, steps, and effects—are referred to as *nodes* of the plan, and are given names so as to distinguish two nodes that have the same content. Finally, plan construction is the process of finding a plan derivation whose yield will achieve a given effect, and plan inference is the process of finding a plan derivation whose yield is a set of observed primitive actions.

[1] Since we assume that the agents have mutual knowledge of the action schemas and that agents can execute surface speech actions, we do not consider beliefs about generation or about the executability of primitive actions.

[2] See the introductory chapter of Allen, Hendler, and Tate (1990) for an overview of planning.

Belief

bel(Agt,Prop): Agt believes that Prop is true.

mb(Agt1,Agt2,Prop): Agt1 and Agt2 mutually believe that Prop is true.

ab(Agt1,Agt2,Prop): Agt1 and Agt2 have an *alternating belief*(Cohen and Levesque, 1990, p. 232) that Prop is true. In other words, either Agt1 believes it, or Agt2 believes it, or Agt1 believes that Agt2 believes it, etc.

Reference

entity(Id,Obj): The discourse entity (Webber, 1983) used to represent the referring expression being built. Id is a unique identifier and Obj is the object being referred to.

ref(Ent,Obj): An action that unifies Obj to the object term of the discourse entity Ent. If the identifier term of Ent is not bound, this action will create a unique identifier for it and will make the value of Obj the referent.

knowref(Agt1,Agt2,Ent): Agt1 knows the referent that Agt2 associates with the discourse entity Ent.

Goals and Plans

goal(Agt,Goal): Agt has the goal Goal. Agents act to make their goals true.

plan(Agt,Plan,Goal): Agt has a plan derivation Plan for achieving Goal. The agent believes that each action contributes to the goal, but not necessarily that all of the constraints hold; in other words, the plan must be coherent (Pollack, 1990, p. 94).

achieve(Plan,Goal): Executing plan will cause Goal to be true. For a goal of knowref, this proposition is true if the plan uniquely identifies the referent (rather than depending on the truth of knowref).

error(Plan,N): Plan has an error at node N. This predicate is used to encode an agent's belief about an invalidity in a plan.

Miscellaneous

subset(Set,Lambda,Subset): Compute the subset, Subset, of Set that satisfies the lambda expression Lambda. This is used as a mental action.

Table 1: Basic Predicates and Actions

3.3 Action Schemas

This section presents action schemas for referring expressions. (We omit discussion of actions that account for superlative adjectives, such as "largest", that describe an object relative to the set of objects that match the rest of the description. A full presentation is given by Heeman (1991).)

As we mentioned, the action for referring, called **refer**, is achieved by surface speech actions. We use decomposition to map **refer** into the surface speech actions, and this decomposition makes use of intermediate actions. Listed below are the actions that we employ and their decomposition into intermediate and surface speech actions (omitting their parameters and mental actions). The symbol $\stackrel{d}{\Longrightarrow}$ may be read as 'decomposes to'.

refer $\stackrel{d}{\Longrightarrow}$ **s-refer describe**
describe $\stackrel{d}{\Longrightarrow}$ **headnoun modifiers**
headnoun $\stackrel{d}{\Longrightarrow}$ **s-attrib**
modifiers $\stackrel{d}{\Longrightarrow}$ { null | **modifier modifiers** }
modifier $\stackrel{d}{\Longrightarrow}$ { **s-attrib** | **s-attrib-rel refer** }

Refer Action

The schema for **refer** is shown in figure 1. (We adopt the Prolog convention that variables begin with an upper-case letter, and all predicates and constants begin with a lower-case letter.) The **refer** action decomposes into two steps: **s-refer**, which expresses the

Header:	refer(Entity)
Where:	speaker(Speaker)
	hearer(Hearer)
Decomposition:	s-refer(Entity)
	describe(Entity)
Effect:	bel(Hearer,goal(Speaker,
	knowref(Hearer,Speaker,Entity)))

Figure 1: **refer** schema

speaker's intention to refer, and **describe**. The variables **Speaker** and **Hearer** are instantiated to **system** or **user**; which is which depends on whether the rule is being used for plan construction or plan inference.

The effect of **refer** is that the hearer should believe that the speaker has a goal of the hearer knowing the referent of the referring expression. The effect has been formulated in this way because we are assuming that when a speaker has a communicative goal she plans to achieve the goal by making the hearer recognize it; the effect will be achieved by the hearer inferring the speaker's plan, regardless of whether or not the hearer is able to determine the actual referent. To simplify our implementation, this is the only effect that is stated for the plan schemas for referring expressions. It corresponds to the literal goal that Appelt and Kronfeld (1987) propose (whereas the actual identification is their condition of satisfaction).

Intermediate Actions

The **describe** action (not shown) is used to construct a description of the object through its decomposition into **headnoun** and **modifiers**. The action **headnoun**, shown in figure 2, has two steps. The first step is the surface speech action **s-attrib**, which determines the

```
Header:          headnoun(Entity,Cand)
Where:           speaker(Speaker)
                 hearer(Hearer)
                 world(World)
Decomposition:   s-attrib(Entity,λX·category(X,Category))
                 subset(World,
                    λX·ab(Speaker,Hearer,category(X,Category)),Cand)
```

Figure 2: **headnoun** schema

head noun of the referring expression and passes back a lambda expression. The second step is the mental action **subset**, which determines the candidate set, **Cand**, associated with the head noun that is chosen. The candidate set is computed by finding the subset of the objects in the world that the speaker believes could be referred to by the head noun—the objects that the speaker and hearer have an appropriate alternating belief about.

Alternating belief is used in order to minimize infelitious reference. Consider the scenario in which the speaker wants to refer to **bird2**, which he believes is mutually believed to be black. Let's also assume that there is another bird that the speaker believes to be brown, but the speaker believes that the hearer believes it is black. By using alternating belief in determining candidate sets, the speaker will find that the description "the black bird" is potentially infelicitous, and will adjust the modifiers accordingly.

The **modifiers** plan (not shown) attempts to ensure that the referring expression that is being constructed is believed by the speaker to allow the hearer to uniquely identify the referent. We have defined **modifiers** as a recursive plan, with two plan schemas. The first schema is used to terminate the recursion, and its constraint specifies that only one object can be in the candidate set. The second schema embodies the recursion. It uses the **modifier** plan, which adds a component to the description and updates the candidate set by computing the subset of it that satisfies the new component. The **modifier** plan thus accounts for individual components of the description. There are two different plan schemas for **modifier**; one is for absolute modifiers, such as "black" and the other is for relative modifiers, such as "larger". We show only the former (figure 3); it decomposes into

```
Header:          modifier(Entity,Cand,NewCand)
Where:           speaker(Speaker)
                 hearer(Hearer)
Decomposition:   s-attrib(Entity,Pred)
                 subset(Cand,λX·ab(Speaker,Hearer,Pred(X)),NewCand)
```

Figure 3: **modifier** schema

the surface speech action **s-attrib** and a mental action that determines the new candidate set, **NewCand**, by including only the objects from the old candidate set, **Cand**, for which the predicate could be believed to be true. The other schema uses the surface speech action `s-attrib-rel` and also includes a step using the top-level plan `refer` to refer to the object of comparison.

Surface Speech Actions

We use three types of surface speech actions. The first is `s-refer`, which is used to express the speaker's intention to refer. The second is `s-attrib`, a set of schemas used for describing an object in terms of an attribute. In figure 4, the schema for describing the color of an object is given. These schemas take as a parameter a lambda expression that encodes the

Header:	`s-attrib(Entity,λX·color(X,Color))`
Where:	`speaker(Speaker)`
	`hearer(Hearer)`
	`ref(Entity,Object)`
Constraint:	`mb(Speaker,Hearer,color(Object,Color))`

Figure 4: An `s-attrib` schema

attribute. The constraint specifies the condition under which the descriptor can be used, which in this case is that the speaker believes that it is mutually believed that the object is of that color. The third type of speech action is `s-attrib-rel`, which is similar to `s-attrib` but, as mentioned earlier, describes an object relative to another object.

3.4 Plan Construction and Plan Inference

The goals that we are interested in achieving are communicative goals. Since these goals cannot be directly achieved by a plan of action, the speaker must instead plan actions that will achieve them indirectly, for instance by planning an utterance that results in the hearer recognizing her goal. So, if the speaker wants to achieve `Goal`, she will attempt to construct a plan whose effect is `bel(Hearer,goal(Speaker,Goal))`.

Plan Construction

Our plan constructor uses a breadth-first search strategy with a heuristic to prune down the search space, so as to achieve a referring expression with the fewest number of actions (cf. E. Reiter, 1990). Given an effect, the plan constructor finds a plan derivation that has a minimal number of primitive actions, that is valid (with respect to the planning agent's beliefs) and whose root action achieves the effect. The yield of this plan derivation can then be given as input to a module that generates the surface form of the utterance. After a plan is constructed, it is added to the speaker's belief space in the form `plan(Speaker,Plan,Goal)`, along with the belief that it achieves the goal.

Plan Inference

Following Pollack (1990), our plan inference process can infer plans in which, in the hearer's view, either a constraint does not hold or a mental action is not executable. In inferring a plan derivation, we first find the set of plan derivations that account for the primitive actions that were observed, without regard to the hearer's beliefs. Second, we evaluate each of these derivations by attempting to find an instantiation for the variables such that all of the constraints hold and the mental actions are satisfiable with respect to the hearer's beliefs about the speaker's beliefs. The plan evaluation process prefers to evaluate them in the order that the plan constructor uses in constructing the plan derivation. However, the plan schemas have been formulated from the perspective of plan construction, and there is a difference in the knowledge that the speaker has when constructing a plan and the knowledge that the hearer has: the speaker knows the goal, the hearer knows only the surface speech actions. So, it might not be efficient or even possible to evaluate the derivations in that order. So, the plan evaluator uses meta-level knowledge to choose the order in which to evaluate the constraints and mental actions in the plan derivation. This knowledge encodes which parameters of a predicate should be instantiated before it can be evaluated.

After the plan evaluation process, if there is just one valid derivation, then the hearer will believe that he has understood. If there is just one derivation and it is invalid, the constraint or mental action that is the source of the invalidity is noted. (We have not explored ambiguous situations, those in which more than one valid derivation remains, or, in the absence of validity, more than one invalid derivation.) From this process, the hearer updates his beliefs to capture the information that was inferred, namely the belief that `plan(Speaker,Plan.Goal)` and a belief about the validity of the plan.

4 Clarifications

4.1 Planning and Clarifying

Clark and Wilkes-Gibbs (1986) have presented a model of how conversational participants collaborate in making a referring action successful (see section 2 above). Their model consists of conversational moves that express a judgment of a referring expression and conversational moves that refashion an expression. However, their model is not computational. They do not account for how the judgment is made, how the judgment affects the refashioning, nor the content of the moves.

Following the work of Litman and Allen (1987) in understanding clarification subdialogues, we formalize the conversational moves of Clark and Wilkes-Gibbs as discourse actions. These discourse actions are meta-actions that take as a parameter a referring expression plan. The constraints and decompositions of the discourse actions encode the conditions under which they can be applied, how the referring expression derivations can be refashioned, and how the speaker's beliefs can be communicated to the hearer. So, the conversational moves, or clarifications[3], can be generated and understood within the planning paradigm.

Surface Speech Actions

An important part of our model is the surface speech actions. These actions serve as the basis for communication between the two agents, and so they must convey the information that is dictated by Clark and Wilkes-Gibbs's model. For the judgment plans, we have the surface speech actions `s-accept`, `s-reject`, and `s-postpone` corresponding to the three possibilities in their model. These take as a parameter the plan that is being judged, and

[3]We use the term *clarification*, since the conversational moves of judging and refashioning a referring expression can be viewed as clarifying it.

for s-reject, also a subset of the speech actions of the referring expression plan. The purpose of this subset is to inform the hearer of the surface speech actions that the speaker found problematic. So, if the referring expression was "the weird creature", and the hearer couldn't identify anything that he thought "weird", he might say "what weird thing", thus indicating he had problems with the surface speech action corresponding to "weird".

For the refashioning plans, we propose that there is a single surface speech action, s-actions, that is used for both replacing a part of a plan, and expanding it. This action takes as a parameter the plan that is being refashioned, and a set of surface speech actions that the speaker wants to incorporate into the referring expression plan. Since there is only one action, if is in uttered in isolation, it will be ambiguous between a replacement and an expansion; however, the speech action resulting from the judgment will provide the proper context to disambiguate its meaning. In fact, during linguistic realization, if the two actions are being uttered by the same person, they could be combined into a single utterance. For instance, the utterance "no, the red one" could be interpreted as a s-reject of the color that was previously used to describe something and an s-expand for the color "red."

So, as we can see, the surface speech actions for clarifications operate on components of the plan that is being built, namely the surface speech actions of referring expression plans. This is consistent with our use of plan derivations to represent utterances. Although we could have viewed the clarification speech actions as acts of informing (cf. Litman and Allen, 1987), this would have shifted the complexity into the parameter of the inform and it is unclear whether anything would have been gained. Instead, we feel that a parser with a model of the discourse and the context can determine the surface speech actions.[4] Additionally, it should be easier for the generator to determine an appropriate surface form.

Judgment Plans

The evaluation of the referring expression plan indicates whether the referring action was successful or not. If it was successful, then the referent has been identified, and so a goal to communicate this is input to the plan constructor. This goal would be achieved by an instance of accept-plan.

If the evaluation wasn't successful, then the goal of communicating the error is given to the plan constructor, where the error is simply represented by the node in the derivation that the evaluation failed at. This goal would either be achieved by an instance of reject-plan or postpone-plan. Now, if the evaluation is not successful, then either no objects match, or more than one matches. In the first case, the referring expression is overconstrained, and the evaluation would have failed on one of the constraints of a surface speech action. In the second case, the referring expression is underconstrained, and so the evaluation would have failed on the constraint that specifies the termination of the addition of modifiers. In our formalization of the conversational moves, we have equated the first case to reject-plan and the second case to postpone-plan, and their constraints test for the abovementioned conditions, by testing for structural properties of where the violation occurred in the plan.

By observing the surface speech action corresponding to the judgment, the hearer, using plan inference, should be able to derive the speaker's judgment plan, and for s-reject and s-postpone, should be able to determine why the speaker found the referring expression plan invalid by evaluating the judgment plan, but without necessarily himself previously

[4]See Levelt (1989, Chapter 12) for how prosody and clue words can be used in determining the type of clarification.

believing the plan to be invalid. This information will provide context for the subsequent refashioning of the referring expression.[5]

Refashioning Plans

If a conversant rejects a referring expression or postpones judgment on it, then either the speaker or the hearer will refashion the expression in the context of the rejection or postponement. In keeping with Clark and Wilkes-Gibbs, we use two discourse plans for refashioning: `replace-plan` and `expand-plan`. The first is used to replace some of the actions in the referring expression plan with new ones, and the second is to add new actions. Replacements can be used if the referring expression either overconstrains or underconstrains the choice of referent, while the expansion can be used only if it underconstrains the choice. So, these plans can check for these conditions.

The decomposition of the refashioning plans encode how a new referring expression can be constructed from the old one. This involves three tasks: first, a single candidate referent is chosen; second, the referring expression is refashioned; and third, this is communicated to the hearer by way of `s-actions`, which was already discussed.[6] The first step involves choosing a candidate. If the speaker of the refashioning is the person who initiated the referring expression, then this choice is obviously pre-determined. Otherwise, the speaker must choose a possible candidate. Goodman (1985) has addressed this problem for the case of when the referring expression overconstrains the choice of referent. He uses heuristics to relax the constraints of the description and to pick one that *nearly* fits it. This problem is beyond the scope of this paper, and so we choose one of the referents arbitrarily (but see Heeman (1991) for how a simplified version of Goodman's algorithm that only relaxes a single constraint can be incorporated into the planning paradigm).

The second step is to refashion the referring expression so that it identifies the candidate chosen in the first step. This is done by using plan repair techniques (Hayes, 1975; Wilensky, 1981; Wilkens, 1985). Our technique is to identify a node in the plan that is an ancestor of the node in error, to construct a replacement for the part of the plan rooted at that node, and then to substitute the replacement into the old plan. This substitution undoes any decisions that were in the removed part that affect other parts of the old derivation. This technique has been encoded into our refashioning plans, and so can be used for both constructing repairs and inferring how another agent has repaired a plan.

Now we consider the effect of these refashioning plans. As we mentioned in section 2, once the refashioning plan is accepted, the common ground of the participants is updated with the new referring expression. So, the effect of the refashioning plans is that the hearer will believe that the speaker wants the new referring expression plan to replace the current one. Note that this effect does not make any claims about whether the new expression will in fact enable the successful identification of the referent. For if it did, and if the new referring expression were invalid, this would imply that the refashioning plan was also invalid, which is contrary to Clark and Wilkes-Gibbs's iterative model of the acceptance process. So, the understanding of a refashioning does not depend on the understanding of the new proposed referring expression, but only on its derivation.

[5]Another approach would be to use this information to revise the beliefs of the participants, so that the refashioning of the plan was influenced by these beliefs rather than the structural properties of where the error occurred. However, such reasoning is beyond the scope of this work.

[6]Another approach would have been to separate the communicative task from the first two (cf. Lambert and Carberry, 1991).

Plan Derivation Predicates

`content(Plan,N,C)`: The node named by `N` has content `C`.

`constraint(Plan,A,C)`: Node `C` is a constraint of action node `A`.

`step(Plan,A,S)`: Node `S` is a step of the action node `A`.

`yield(Plan,A,Y)`: Node `A` has a yield of the primitive actions `Y`.

Plan Repair Actions

`construct(Goal,Plan,Actions)`: Construct a plan that achieves `Goal`. `Actions` are the primitive actions of the constructed plan.

`substitute(Plan,Node,NewPart,NewPlan,NewActions)`: Undo all variable bindings in `Plan` (except those in primitive actions that are not object terms of discourse entities), and then substitute the content of `Node` in `Plan` by `NewPart`. The result of this is the plan `NewPlan` and the new primitive actions `NewActions`.

`evaluate(Plan)`: Evaluate `Plan`; succeed only if the plan is valid. (This is treated as a mental action, in order to avoid the use of *post-constraints*.)

Plan Replacement

`replace(Plan,NewPlan)`: The plan derivation `NewPlan` replaces `Plan`.

Table 2: Predicates and Actions

The distinction between the effect of the refashioning plans and the effect of a referring action itself relates to Grosz and Sidner's work (1986) on intention and discourse structure. The refashionings are discourse segments embedded within the discourse segment of the referring action; this corresponds to the intention of the refashionings being dominated by the intention of the referring action. But, the intentions of the refashionings are not in a dominance relationship with one another; they are all at the same level in the discourse structure.

4.2 Notation for Action Schemas

Before presenting the action schemas for clarifications, we need to introduce the notation that these schemas will use. This notation is motivated by work of Litman and Allen (1987) in understanding clarification subdialogues. The first four predicates in Table 2 are for reasoning about the structural properties of a plan derivation. The next three are actions used for refashioning a plan, and the last is for representing that one plan is a replacement of another. Throughout the table, `Plan` refers to a plan derivation.

4.3 Action Schemas

This section presents plan schemas for clarifications. To simplify our implementation, the surface speech actions have been stated without any effects or constraints.

accept-plan

The discourse action `accept-plan`, shown in figure 5, is used by the speaker to establish the mutual belief that a plan will achieve its goal. The constraints of the schema specify

```
Header:         accept-plan(Plan)
Where:          speaker(Speaker)
                hearer(Hearer)
Constraint:     achieve(Plan,Goal)
Decomposition:  s-accept(Plan)
Effect:         bel(Hearer,goal(Speaker,mb(Speaker,Hearer,
                    achieve(Plan,Goal))))
```

Figure 5: `accept-plan` schema

that the plan being accepted achieves its goal and the decomposition is the surface speech action `s-accept`. The effect of the schema is that the hearer will believe that the speaker has the goal that it be mutually believed that the plan achieves its goal.

reject-plan

The discourse action `reject-plan`, shown in figure 6, is used by the speaker if the referring expression plan overconstrains the choice of referent. The speaker uses this schema in order

```
Header:         reject-plan(Plan)
Where:          speaker(Speaker)
                hearer(Hearer)
Constraint:     error(Plan,ErrorNode)
                constraint(Plan,ParentPlan,ErrorNode)
                yield(Plan,ParentPlan,Acts)
                length(Acts,1)
Decomposition:  s-reject(Plan,Acts)
Effect:         bel(Hearer,goal(Speaker,mb(Speaker,Hearer,
                    error(Plan,ErrorNode))))
```

Figure 6: `reject-plan` schema

to tell the hearer that the plan is invalid and which node the evaluation failed at. The constraints require that the error occurred at a constraint of a surface speech action. The constraints first determine the node, `ErrorNode`, in the derivation that the evaluation failed at. Second, they ensure that `ErrorNode` is a constraint of some plan instance, `ParentPlan`. Third, they check that the yield of `ParentPlan` consists of only a single surface speech action. The decomposition consists of `s-reject`, which takes as its parameter the surface speech action that was determined to be part of the cause of the error.

postpone-plan

The schema for `postpone-plan` (not shown) is similar to `reject-plan`. However, it requires that the error in the evaluation occurred at the constraint of the instance of `modifiers` that has a null decomposition—in other words, the `modifiers` instance that terminates the addition of modifiers.

replace-plan

The **replace-plan** schema, shown in figure 7, is used by the speaker to replace some of the primitive actions in a plan with new actions. Its constraints require that the error occurred

Header:	`replace-plan(Plan)`
Where:	`speaker(Speaker)`
	`hearer(Hearer)`
Constraint:	`error(Plan,ErrorNode)`
	`constraint(Plan,ParentNode,ErrorNode)`
	`step(Plan,ModifierNode,ParentNode)`
	`content(Plan,ModifierNode,ModifierContent)`
	`ModifierContent = modifier(Entity,Cand,Cand1)`
Decomposition:	`member(Object,Cand)`
	`ref(Entity,Object)`
	`construct(modifier(Entity,Cand,Cand1),Replacement,Acts)`
	`substitute(Plan,ModifierNode,Replacement,NewPlan,Acts)`
	`evaluate(NewPlan)`
	`s-actions(Plan,Act)`
Effect:	`bel(Hearer,goal(Speaker,mb(Speaker,Hearer,`
	`replace(Plan,NewPlan))))`

Figure 7: **replace-plan** schema

at a constraint, `ErrorNode`, of a subplan, `ParentNode`, that is a step of `modifier`—in other words, the error occurred at the constraint of a surface speech action. As one can see, the formulation of these constraints is somewhat awkward, and does not capture the case in which the violation occurred on the surface speech action that **headnoun** decomposes into. The reason for this is that the constraints serve the additional function of extracting information that will be needed by the steps of the decomposition, namely `Cand`, `Entity`, and `ModifierNode`.

The decomposition of the schema specifies how a new referring expression plan can be built. The first step, `member(Object,Cand)`, chooses one of the objects that matched the part of the description that preceded the error; if the speaker is not the initiator of the referring expression, then this is an arbitrary choice. The second step maps the chosen object to the discourse entity. The third step, through a recursive call to the plan constructor, builds a replacement for the `modifier` subplan that was identified in the constraints. This replacement will distinguish the chosen candidate from the rest. The fourth step substitutes the replacement into the current referring expression, resulting in the refashioned referring expression `NewPlan`. The fifth step, through a call to the plan evaluator, ensures that `NewPlan` actually identifies a unique object. This is necessary, because the step that chooses the candidate does not consider the constraints imposed by the surface speech actions that follow the one in error, and also, the step that builds the replacement is ignorant of how the replacement will interact with the rest of the description. Finally, the sixth step is the surface speech action `s-actions`, which is used to inform the hearer of the surface speech actions that are being added to the referring expression plan.

expand-plan

The **expand-plan** schema (not shown) is similar to **replace-plan**. The difference is that instead of replacing some of the primitive actions, it replaces the terminal instance of **modifiers** by a **modifiers** subplan that distinguishes one of the objects from the others that match, thus effecting an expansion of the surface speech actions.

4.4 Plan Construction and Plan Inference

The general plan construction and plan inference processes are essentially the same as those for referring expressions. However, the plan inference process has been augmented so as to embody the criteria for understanding that were outlined in section 4.1. The inference of judgment plans must be sensitive to the fact that such a plan includes the constraint that the speaker found the judged plan to be in error even though the hearer might not believe it to be. So, the inference process is allowed to assume that the speaker believes any constraint that the goal of the plan implies.

In the case of a refashioning, the hearer might not view the proposed referring expression plan as being sufficient for identifying the referent, but would nonetheless understand the refashioning. So, the inference process requires only that the proposed referring expression be derived—so that it can serve to replace the current plan—but not that it be acceptable. This has been effected by giving a special meaning to the mental actions `construct` and `evaluate`. When a `construct` is inferred, the plan that is a parameter of `construct` is derived but not evaluated. Likewise for `evaluate`, its parameter, a plan, is not evaluated.[7]

5 Modeling Collaboration

In the last two sections, we discussed how initial referring expressions, judgments, and refashionings can be generated and understood in our plan-based model. In this section, we show how plan construction and plan inference fit into a complete model of how an agent collaborates in making a referring action successful. Previous natural language systems that use plans to account for the surface speech acts underlying an utterance (such as Cohen and Perrault, 1979; Allen and Perrault, 1980; Appelt, 1985; Litman and Allen 1987) model only the recognition or only the construction of an agent's plans, and so do not address this issue.

In a dialogue, the goals that a speaker plans to achieve are influenced by the plans that she has attributed to her conversational partner. This influence is a change in the mental state of the participant. We model this by using *acceptance rules* and *goal adoption rules*. The term "acceptance rule" is motivated by the work of Clark and Schaefer (1989) on contributing to discourse. Contributions are subjected to an acceptance process, and once they are accepted, the common ground of the participants is updated. So, our acceptance rules state the conditions under which a contribution is accepted, the result being that the beliefs of the agent are updated. These acceptance rules are used not only by the hearer, but also by the speaker to reflect her own contribution to the common ground.[8] Our other

[7]Another approach would be to have the plan inference process reason about the intended effects of the plan that it is inferring in order to decide whether it should evaluate embedded plans and whether this evaluation should affect the evaluation of the parent plan.

[8]A question that we have not addressed is when these rules should be applied. We currently assume that the speaker presupposes the hearer's acceptance of the plan underlying an utterance.

rules, goal adoption rules, give the conditions under which a goal can be adopted.

These rules, however, give us only a partial account of collaborative activity. The goals that agents adopt do not just arise from the other participant's utterances, but are due to what Clark and Wilkes-Gibbs refer to as a *mutual responsibility* for the success of a referring action, or what Searle (1990) refers to as a *we-intention*. This allows the agents to interact so that neither assumes control of the dialogue, thus allowing both to contribute to the best of their ability without being guided or impeded by the other. This is different from what Grosz and Sidner (1990) have called master-servant dialogues, which occur in teacher-apprentice or information-seeking dialogues, in which one of the participants is controlling the conversation (cf. Walker and Whittaker, 1990). Note that the non-controlling agent may be helpful by anticipating obstacles in the plan (Allen and Perrault, 1980), but this is not the same as collaborating.

The question now arises as to how the state of an agent who is engaged in a collaborative activity should be modeled. We propose that in addition to an intention to achieve some goal, which in our case is to refer, the agents also have a plan that they are currently considering in order to achieve the goal. This plan serves to coordinate their activity and so agents will have intentions to keep this plan in their common ground. The plan need not be valid (unlike the *shared plan* of Grosz and Sidner (1990)), so the agents might not mutually believe that each action contributes to the goal of the plan. Since the plan might be invalid, agents will have a belief regarding the validity of the plan, and an intention that this belief be mutually believed.

The discourse plans that we described in the previous section can now be seen as plans that can be used to further the collaborative activity. Judgment plans express beliefs about the success of the current plan, and refashioning plans update it. So, the mental state of an agent sanctions the adoption both of goals to express judgment and of goals to refashion, and it sanctions the acceptance of these plans and so the updating of beliefs about the current plan.[9]

In section 4.1, we discussed conditions under which an agent could be viewed as understanding a judgment or refashioning plan. For a judgment, it was that the hearer know which constraint the speaker found in error, but not necessarily to agree with the error. For a refashioning, it was to recognize the proposed referring expression plan, but not necessarily to find it acceptable. Now, we need to examine the criteria for accepting these plans. Remember that the agents are engaged in a collaborative activity, and so they have an intention both to achieve the goal underlying this activity and to coordinate their activity. We propose that this results in the agents always accepting these plans so long as they are understood. For a judgment plan, this is reasonable, since although the hearer might not agree with the suggestion of error, he should realize that the referring expression must be mutually acceptable in order for the identification to properly take place. For a refashioning, this also is reasonable, for if he doesn't find the resulting referring expression adequate, he can still accept it and then proceed to refashion it. This is simpler than the alternative, which is to reject the speaker's refashioning, and trying to refashion that.

[9]The collaborative activity also sanctions discourse expectations that the other participant's utterances will pertain to the collaborative activity. We do not explicitly address this however.

5.1 Rules

Now that we have outlined our model, we can give the rules that our system uses.[10] These rules have been revised from an earlier version (Heeman, 1991) so as to better model the acceptance process. Like their predecessors, these rules embody the assumption that judgment and refashioning plans are always understood. This is evidenced through the rules not checking the validity of these plans and having no means to repair them.

Entering into a Collaborative Activity

We need a rule that permits an agent to enter into a collaborative activity. We use the predicate **cstate** to represent that an agent is in such a state, and this predicate takes as its parameters the agents involved, the goal they are trying to achieve, and their current plan. Our view of how such a collaborative activity can be entered is very simple: if the agent has constructed or inferred a referring expression plan, then it enters into a collaborative activity, as shown below:[11]

```
cstate(Speaker,Hearer,CPlan,Goal) ⟸
     plan(Speaker,CPlan,Goal) &
     Goal = knowref(Hearer,Speaker,Entity)
```

Acceptance Rules

In order to model how the state of the collaborative activity progresses, we need an acceptance rule for each type of utterance that will be contributed. As mentioned earlier, these rules are used by both the hearer and speaker of the utterance. So, in particular, it is these rules that sanction the speaker of a refashioning to update the current plan to be the plan that she is proposing.

The first acceptance rule, given below, is used to accept a judgment plan, JPlan, whose goal is to make it mutually believed that there is an error in the current plan, CPlan, that corresponds to a collaborative activity. The application of this acceptance rule causes the participant applying it to adopt the belief that it is mutually believed that there is an error in the plan,[12] which in turn causes the retraction of any beliefs that it achieves the goal.

```
mb(Speaker,Hearer,error(CPlan,Node)) ⟸
     cstate(Speaker,Hearer,CPlan,Goal) &
     plan(Speaker,JPlan,mb(Hearer,Speaker,error(CPlan,Node)))
```

The second rule is similar to the first, except that it is concerned with accepting refashionings. The application of the rule causes the participant applying it to update his common ground, in other words, to update the current plan with the one being proposed. So, in actuality, this rule is about belief revision. Our belief module, when given this belief, will

[10] For simplicity, we represent the rules for entering into a collaborative activity, adopting beliefs, and adopting goals with the same operator, $\Longleftarrow$. For a more formal account, three different operators should be used.

[11] The rules also include the predicates speaker(Speaker) and hearer(Hearer) to instantiate the variables Speaker and Hearer.

[12] To simplify our belief module, we model the adoption of a mutual belief as just the adoption that the agent believes it and that he believes the other participant believes it.

update the cstate by replacing the current plan, CPlan, with NewPlan, and will evaluate NewPlan to determine whether it is valid.

```
mb(Speaker,Hearer,replace(CPlan,NewPlan)) ⟸
     cstate(Speaker,Hearer,CPlan,Goal) &
     plan(Speaker,RPlan,mb(Speaker,Hearer,replace(CPlan,NewPlan)))
```

The third rule is for accepting a judgment plan that accepts the current plan. This rule can only be applied if the participant believes that the current plan achieves the goal.

```
mb(Speaker,Hearer,achieve(CPlan,Goal)) ⟸
     cstate(Speaker,Hearer,CPlan,Goal) &
     achieve(CPlan,Goal) &
     plan(Speaker,JPlan,mb(Speaker,Hearer,achieve(CPlan,Goal)))
```

Adopting Goals

The next set of rules captures how an agent adopts goals in order to collaborate in achieving the goal of the activity. We refer to the agent who is adopting a goal as the speaker.

The first rule, given below, is used to adopt the goal of informing the hearer that there is an error in CPlan. The conditions specify that CPlan is the current plan of a collaborative activity, that there is an error in the plan, and that this is not already mutually believed.[13]

```
goal(Speaker,mb(Speaker,Hearer,error(CPlan,Node))) ⟸
     cstate(Speaker,Hearer,CPlan,Goal) &
     error(CPlan,Node) &
     not(mb(Speaker,Hearer,error(CPlan,Node)))
```

The second rule is used to adopt the goal of replacing the current plan, CPlan, if it has an error. It is similar to the first rule, but it requires that the speaker believe that it is mutually believed that there is an error in the current plan. So, this goal cannot be adopted before the goal of expressing judgment has been planned.

```
goal(Speaker,mb(Speaker,Hearer,replace(CPlan,NewPlan))) ⟸
     cstate(Speaker,Hearer,CPlan,Goal) &
     mb(Speaker,Hearer,error(CPlan,Node))
```

The third rule is used to adopt the goal of communicating the speaker's acceptance of the current plan.

```
goal(Speaker,mb(Speaker,Hearer,achieve(CPlan,Goal))) ⟸
     cstate(Speaker,Hearer,CPlan,Goal) &
     achieve(CPlan,Goal) &
     not(mb(Speaker,Hearer,achieve(CPlan,Goal)))
```

[13] The not mb on the third condition means that the speaker has no evidence that it is mutually believed, which is the *negation-by-failure* approach.

5.2 Applying the Rules

The rules that we gave are used to update the mental state of the agent and to guide its activity. Acting as the hearer, the system performs plan inference on each set of actions that it observes, and then applies any acceptance rule or collaborative activity rule that it can. When all of the observed actions are processed, the system switches from the role of hearer to speaker.

As the speaker, the system checks the rules to find a goal that it can adopt, and then constructs a plan to achieve it. Next, presupposing the other participant's acceptance of the plan, it applies any acceptance rule or collaborative activity rule that it can. It repeats this until there are no more goals to adopt. One exception is that a goal to make it mutually believed that a plan achieves a goal cannot be in the same response as the proposal of that plan! The actions of the constructed plans form the response of the system; in a complete natural language system, they would be converted to a surface utterance. The system then switches to the role of hearer.

6 An Example

We are now ready to illustrate our system in action.[14] For this example, we use a simplified version of a subdialogue from the London-Lund corpus (Svartvik and Quirk, 1980, S.2.4a:1–8):

(6.1) **A:** [1] See the weird creature.
B: [2] In the corner?
A: [3] No, on the television.
B: [4] Okay.

The system will take the role of person B and we will give it the belief that there are two objects that are "weird"—a television antenna, which is on the television, and a fern plant, which is in the corner.

6.1 Understanding "The weird creature"

For the first sentence, the system is given as input the surface speech actions underlying "the weird creature," as shown below:

```
s-refer(Entity)
s-attrib(Entity,λX·assessment(X,weird))
s-attrib(Entity,λX·category(X,creature))
```

The system invokes the plan inference process, which first finds a plan derivation whose yield is the above set of surface speech actions. This results in the plan derivation shown in figure 8; arrows represent decomposition, and for brevity, constraints and mental actions have been omitted and the parameters only of the surface speech actions are shown.

[14]The system is implemented in C-Prolog under Unix.

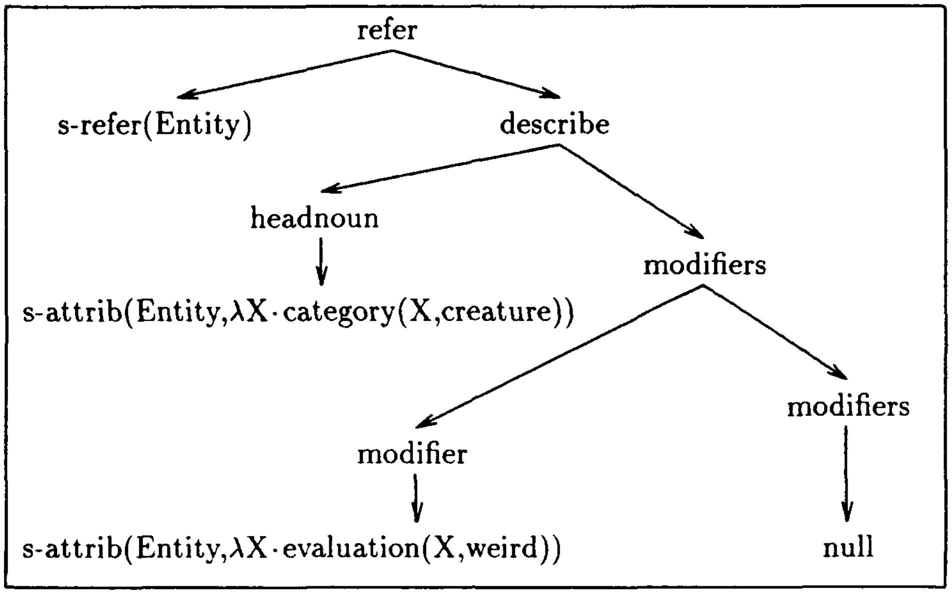

Figure 8: Plan derivation for "The weird creature"

Next, the plan derivation is evaluated. The **subset** action in the **headnoun** plan is evaluated first, which narrows the candidate set to the antenna and the fern plant. The **subset** action in the **modifier** plan is then evaluated, which does not eliminate either of the candidates, since the system finds both of them "weird." The constraint on the **modifiers** plan that terminates the addition of modifiers is then evaluated. However, this constraint fails, since there are two objects that match the description rather than one, as required. The system adds the plan derivation to its belief space, and the belief that it failed on this constraint.

Now that the plan inference process is finished, the system tries to update its mental state. This leads to the system entering into a collaborative activity, in which the goal is for it to know the referent. The current plan for this is the plan that was just inferred.

6.2 Constructing "In the corner?"

The system next checks whether there are any goals that it should adopt. Since the current plan of the collaborative activity is problematic, the system gives itself the goal of making this belief mutually believed. Since the referring expression is underconstrained, the plan constructor builds an instance of **postpone-plan**. The system then applies the acceptance rule to adopt the belief that it is mutually believed that there is an error in the plan, and so presupposes the user's acceptance of the judgment plan.

The system next checks to see whether there are any other goals it should adopt. This leads it to adopting the goal of refashioning the invalid referring expression plan and of informing the user of the new plan. To achieve this goal, the plan constructor builds an instance of **expand-plan**. In doing this, the system chooses one of the objects that matched the original description as the likely referent; in this case it happens to choose the object in the corner. It then constructs an expansion to distinguish this object from the others that matched the description, and this expansion, "in the corner," is incorporated into the old referring expression plan, thereby creating a new expanded plan. The new plan is shown in figure 9, with the expansion circled (we have abbreviated the derivation of "the corner"). The surface speech action of **expand-plan** is `s-actions`, which takes the surface speech actions of the expansion as its parameter.

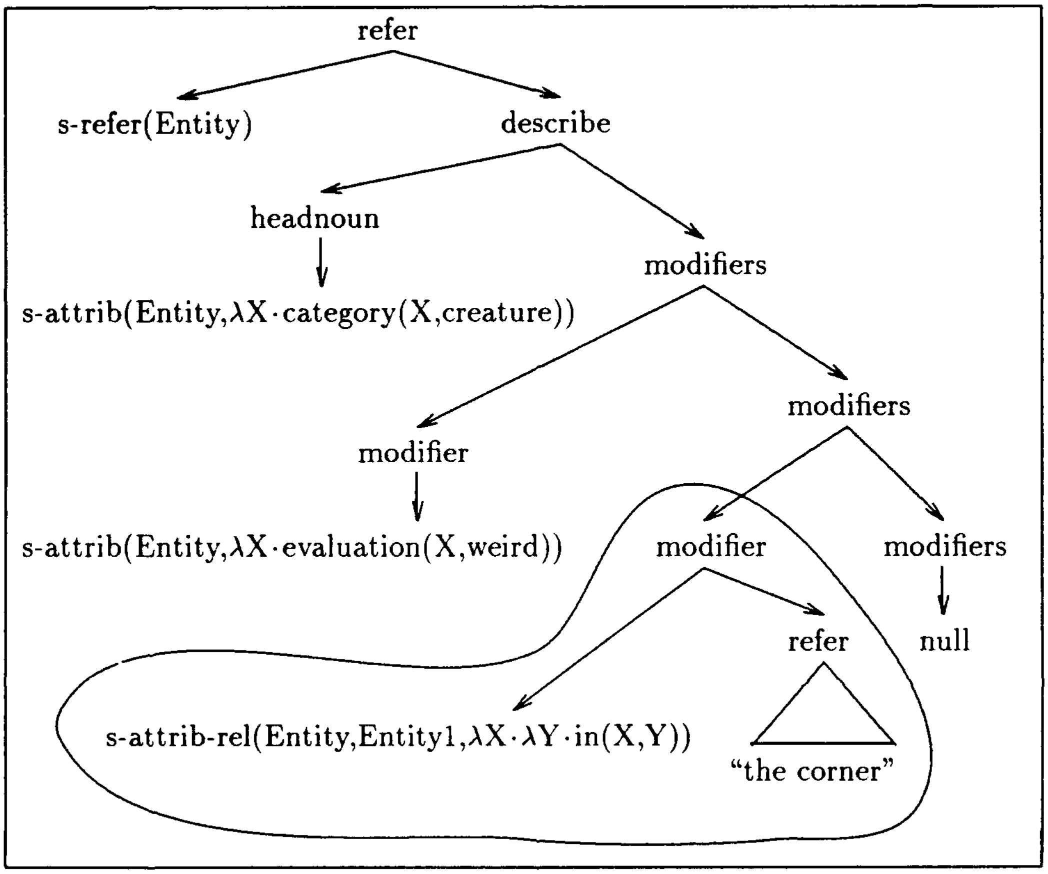

Figure 9: Plan derivation for "The weird creature in the corner"

Next, the system applies the acceptance rule corresponding to a refashioning, and so adds the belief that the new expanded plan replaces the old referring expression plan. This causes the belief module to update the current plan of the collaborative activity, and to add the belief that the new plan achieves the goal.

The two plans that were constructed, **postpone-plan** and **expand-plan**, give rise to the output of the surface speech actions **s-postpone** and **s-expand**, which would be realized as "in the corner?"

6.3 Understanding "No, on the television"

The user next utters "No, on the television." This would get parsed into two separate surface speech actions, an `s-reject` corresponding to "no", and an `s-actions` corresponding to "on the television." For simplicity, the plan inference process is invoked separately on each.

The system starts with the `s-reject` action. We assume that the parser can determine from context that the "no" is rejecting that the referent is "in" something, and so the parameter of s-reject is the s-attrib-rel action. From this, it derives a plan whose yield is the s-reject action, and this plan is an instance of reject-plan. The system then evaluates the constraints and mental actions of the plan, which results in it determining which constraint the user found to be in error. In this case, it is the constraint associated with the surface speech action s-attrib-rel, that it is mutually believed that there is a weird creature that is *in* something.

The system then applies the appropriate acceptance rule, and so adds the mutual belief that there is an error in the current plan. With this belief, the system will have the context that it needs to understand the user's refashioning plan.

The system next performs plan recognition starting with the second surface speech action, s-actions, which corresponds to the refashioning "on the television". So, it takes as a parameter the following list of actions:[15]

```
s-attrib-rel(Entity,Entity2,λX·λY·on(X,Y))
s-refer(Entity2)
s-attrib(Entity2,λX·category(X,television))
```

The system finds two plan derivations that account for the primitive action, one an instance of replace-plan and the other an instance of expand-plan. Next it evaluates the constraints and mental actions. This allows it to eliminate the instance of expand-plan, since the constraint that the error occurred on the terminating instance of modifiers is not satisfiable. The system is able to successfully evaluate the instance of replace-plan. In doing this, it derives the replacement that the user is proposing, and it substitutes this into the current referring expression, so giving the proposed referring expression; however, the proposed expression is not evaluated at this point. Figure 10 shows the new expression, with the replacement circled.

The system then applies the acceptance rule for refashioning plans, and so adds the belief that it is mutually believed that the new referring expression plan replaces the old plan. This causes the belief module to update the current plan, and to evaluate it. The subplan corresponding to "the television" is understood without problem,[16] and the modifier corresponding to "on the television" is able to narrow down the candidates that matched "weird creature" to a single object. So, the new current plan is found to be valid, and the system adds the belief that the plan achieves the referring action, which is the goal it is collaborating upon.

6.4 Constructing "Okay"

Since the system believes that the plan achieves the goal of the collaborative activity, it adopts the goal of informing the user of this. The plan constructor achieves this by planning an instance of accept-plan, which results in the surface speech action s-accept, which would be realized as "Okay." The system then applies an acceptance rule, and so adopts the belief that it is mutually believed that the plan achieves the goal of referring.

[15] We assume that the parser determines the appropriate discourse entities in these actions: Entity is the discourse entity for the object being referred to, and that Entity2 is different from it.

[16] If "the television" is not understood, then since it is a referring expression in its own right, the conversants could collaborate on identifying its referent independently of the referent of "the weird creature;" that is the participants could enter into an embedded collaborative activity by focusing on one part of the current plan.

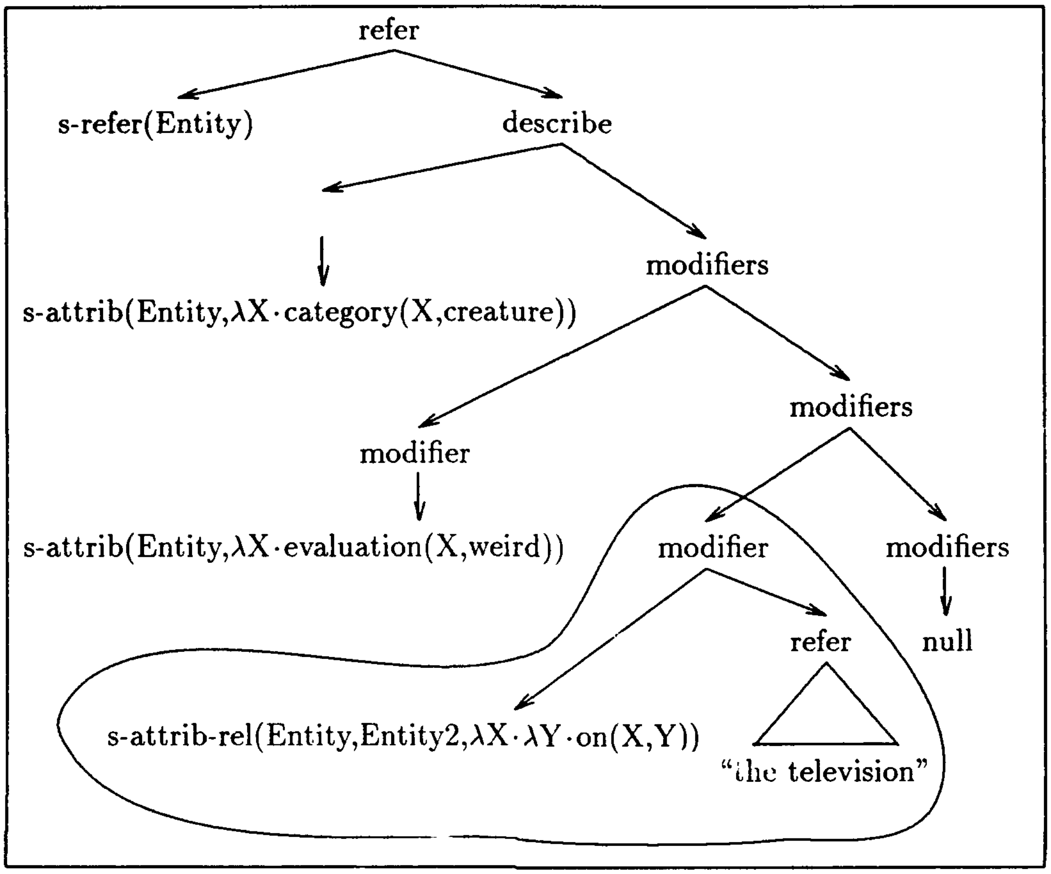

Figure 10: The plan derivation for "The weird creature on the television"

7 Comparisons to Related Work

In providing a computational model of how agents collaborate upon referring expressions, we have touched on several different areas of research. First, our work has built on previous work in referring expressions, especially their incorporation into a model based on the planning paradigm. Second, our work has built on the research done in modeling clarifications in the planning paradigm and on plan repair. Third, our work is related to the research being done on modeling collaborative and joint activity.

7.1 Referring Expressions

Cohen (1981) and Appelt (1985) have also addressed the generation of referring expressions in the planning paradigm. They have integrated this into a model of generating utterances, a step that we haven't taken. However, we have extended their model by incorporating even the generation of the components of the description into our planning model. One result of this is that our surface speech actions are much more fine-grained.

7.2 Clarifications and Plan Repair

An important part of our work involves accounting for clarifications of referring expressions by using meta-actions that incorporate plan repair techniques. This approach is based on Litman and Allen's work (1987) on understanding clarification subdialogues, in which meta-actions were used to model discourse relations, such as clarifications. There are several major differences between our work and theirs. First, our work addresses not only understanding but also generation and how these two tasks fit into a model of how agents collaborate in discourse. Second, Litman and Allen use a stack of unchanging plans to represent the state of the discourse. We, however, use a single *current plan*, modifying it as clarifications are made. This difference has an important ramification, for it results in different interpretations of the discourse structure. Consider dialogue (7.1), which was collected at an information booth in a Toronto train station (Horrigan, 1977). (Although the participants are not collaborating in making a referring expression, the dialogue will serve to illustrate our point.)

(7.1) **P:** [1] The 8:50 to Montreal?
C: [2] 8:50 to Montreal. Gate 7.
P: [3] Where is it?
C: [4] Down this way to your left. Second one on the left.
P: [5] OK. Thank you.

Litman and Allen represent the state of the discourse after the second utterance as a clarification of the passenger's `take-train-trip` plan. The information that the train boards at gate 7 is represented only in the clarification plan. So, when the passenger asks "Where is it?", their system, acting as the clerk, cannot interpret this as a clarification of the `take-train-trip` plan, since the utterance "cannot be seen as a step of [that] plan" (p. 188). So, it is interpreted instead as a request for a clarification of the clerk's "Gate 7" response, implicitly assuming that "Gate 7" was not accepted. In our model, the acceptance of "Gate 7" would be presupposed, and so it would be incorporated into the `take-train-trip` plan. So, the passenger's question of "Where is it?" would be viewed as a request for the clerk to clarify that plan.

The work of Moore and Swartout (1991), Cawsey (1991), and Carletta (1991) on interactive explanations also addresses clarifications using plan repair techniques. This body of work uses plan construction techniques to generate explanations, and uses the constructed plan as a basis for recovery strategies if the user doesn't understand the explanation. In the cases of Cawsey and Carletta, both use meta-actions to encode the plan repair techniques.

Other relevant work is that of Lambert and Carberry (1991). In their model of understanding information-seeking dialogues, they propose a distinction between problem-solving activities and discourse activities. In contrast, our clarifications embody both functions in the same actions, thus allowing for a simpler approach to inferring the refashioned referring expressions, since we need not chain to a meta-operator.

7.3 Collaboration

Grosz, Sidner, and Lochbaum (Grosz and Sidner, 1990; Lochbaum, Grosz and Sidner, 1990) are interested in the type of plans that underlie discourse in which the agents are collaborating in order to achieve some goal. They propose that agents are building a *shared plan* in which participants have a collection of beliefs and intentions about the actions in the plan. Our model differs from theirs in two important aspects. First, not only do agents have a collection of beliefs and intentions regarding the actions of a shared plan, we feel that they also have an intention about the goal (Searle, 1990; Cohen and Levesque, 1991). It is this intention, in conjunction with the current plan, that sanctions the adoption of beliefs and intentions about potential actions that will contribute to the goal, rather than just the shared plan.

Second, we feel that their definition of a partial shared plan is too restrictive. Although they address partial beliefs, they require, in order for an action to be part of a partial shared plan, that both agents believe that the action *contributes* to the goal. However, this is too strong. In collaborating to achieve a mutual goal, participants sometimes propose an action that is not believed by the other participant or even by the participant that is proposing it. In failing to represent such states, their model is unable to represent the intermediate states in which a hearer might have understood how the speaker's utterance contributes to a plan, but doesn't agree with it. This is important, since if the refashioned plan is invalid, only the referring expression should be refashioned, not the refashioning itself.

Cohen and Levesque (1991) focus on formalizing joint intention in a logic. They use this formalism to explain how such elements of communication as confirmations arise when agents are engaging in a joint action. However, they have not addressed how agents collaborate in building a plan, only how agents collaborate while executing a plan. Once this limitation is overcome, their approach could offer us a route for formalizing the mental states of the collaborating agents in our model and for proving that our acceptance and goal adoption rules follow from such states.

Traum (1991) is concerned with reaching mutual understanding in dialogues. So far, Traum has focused on the speech actions that are needed, and he proposes speech actions for controlling turn-taking and grounding, in addition to such speech actions as informing, suggesting, accepting a domain plan, and rejecting a domain plan. In representing the current state of a dialogue, Traum proposes a number of different plan spaces, corresponding to whether a plan (or action) is just privately held, or has been proposed, acknowledged, or accepted. Our work has assumed a simpler model of both the speech actions and the mental state of an agent: agents do not reason about the plan in advance of making a contribution, acknowledgements are presupposed, and the acceptability of the actions in a plan is modeled by a belief about the validity of the plan. However, by concentrating on referring expressions, and by making a number of simplifications, we have been able to investigate the link between the speech actions and the mental state of an agent during a collaborative activity.

8 Conclusion

We have presented a computational model of how a conversational participant collaborates in making and understanding a referring expression, based on the view that language is goal-oriented behavior. This has allowed us to do the following. First, we have accounted for the tasks of building a referring expression and identifying its referent by using plan construction and plan inference. Second, we have accounted for the conversational moves that participants make during the acceptance process by using meta-actions. Third, we have accounted for collaborative activity by proposing that agents are in a certain mental state that includes a goal, a plan that they are currently considering, and intentions. This mental state sanctions the acceptance of clarification plans, and sanctions the adoption of goals to clarify. Although our work has focused on referring expressions, we feel that it is relevant to collaboration in general and to how agents contribute to discourse.

This paper is based on the work of Clark and Wilkes-Gibbs (1986). We have proposed speech acts for judging and refashioning a referring expression, and shown how these speech acts can be generated and understood in the planning paradigm, and how they relate to the participants' mutual responsibility. Thus, we have taken their descriptive model of the collaborative process and recast it into a computational model, demonstrating the computational feasibility of their model and its compatibility with current practices in artificial intelligence.

There are many ways that this research could be extended. Perhaps the most obvious would be to extend the planning component of our model. First, our coverage of referring expressions could be extended to handle references to objects in focus and to descriptions that include a plan of physical actions for identifying the referent. Second, the treatment of clarifications could be improved; specifically, how plan failures are reasoned about, how plan failures affect the agent's beliefs, and how these failures are repaired. Third, this research needs to be integrated into a more complete plan-based approach to language, and needs to be extended so as to handle more general discourse plan failures (McRoy and Hirst, 1991; Horton and Hirst, 1991). ` benchmark for such future work could be dialogue (8.1) below, from the London-Lund corpus (Svartvik and Quirk, 1980, S.2.4a:1–8), which is the basis of the example used in section 6. This dialogue shows how collaboration on a referring expression can be embedded in other activities, how agents can return back to a collaborative activity, and even how agents can take advantage of a mistaken referent.

(8.1) **A:** [1] What's that weird creature over there?
B: [2] In the corner?
A: [3] *affirmative noise*
B: [4] It's just a fern plant.
A: [5] No, the one to the left of it.
B: [6] That's the television aerial. It pulls out.

A second avenue for future work is to further investigate collaborative behavior and protocols for interaction. We need to formalize what it means for agents to be collaborating, in a theory that takes account of rational interaction and the beliefs and knowledge of the participants. Such a theory would do the following. First, it would give a more complete motivation for the processing rules that we used for how agents interact in a collaborative activity. Second, it would account for why agents would enter into such a mode of interaction, how it is initiated, how it is carried forward (especially how agents' beliefs and knowledge influence their actions), and how it ends. Third, it would be extendable to other forms of interaction, such as information-seeking dialogues. Fourth, it would specify how collaborative activity could be embedded in, or embed, other types of interactions. By answering these questions, we will not only have a better model to base natural language interfaces on, but we will also have a better understanding of how people interact.

Acknowledgments

We would like to thank James Allen and Hector Levesque for their comments on this paper. We would also like to especially thank Janyce Wiebe for her invaluable contribution to an earlier version of this work. As well, we are grateful for comments from, and discussions with, Diane Horton, Susan McRoy, Massimo Poesio, and David Traum. Funding at the University of Toronto and the University of Rochester was provided by the Natural Sciences and Engineering Research Council of Canada, with additional funding at Rochester provided by NSF under Grant IRI-90-13160 and ONR/DARPA under Grant N00014-92-J-1512.

References

Allen, J., Hendler, J., and Tate, A., editors (1990). *Readings in Planning.* Morgan Kaufmann Publishers.

Allen, J. F. and Perrault, C. R. (1980). Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178. Reprinted in (Grosz, Sparck Jones and Webber, 1986).

Appelt, D. and Kronfeld, A. (1987). A computational model of referring. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '87)*, pages 640–647.

Appelt, D. E. (1985). Planning English referring expressions. *Artificial Intelligence*, 26(1):1–33.

Austin, J. L. (1962). *How to do things with words.* Oxford University Press, New York.

Carletta, J. (1991). Recovering from plan failure using a layered architecture. Research Paper 524, Department of Artificial Intelligence, University of Edinburgh.

Cawsey, A. (1991). Generating interactive explanations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '91)*, pages 86–91.

Clark, H. H. and Marshall, C. R. (1981). Definite reference and mutual knowledge. In Joshi, A. K., Webber, B. L., and Sag, I., editors, *Elements of Discourse Understanding*, pages 10–62. Cambridge University Press, Cambridge.

Clark, H. H. and Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science*, 13:259–294.

Clark, H. H. and Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition*, 22:1–39.

Cohen, P. R. (1981). The need for referent identification as a planned action. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 31–36.

Cohen, P. R. and Levesque, H. J. (1990). Rational interaction as the basis for communication. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, SDF Benchmark Series, pages 221–255. MIT Press.

Cohen, P. R. and Levesque, H. J. (1991). Confirmation and joint action. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '91).*

Cohen, P. R. and Perrault, C. R. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212. Reprinted in (Grosz, Sparck Jones and Webber, 1986).

Dale, R. (1989). Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 68–75.

Goodman, B. A. (1985). Repairing reference identification failures by relaxation. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 204–217.

Grosz, B. J. and Sidner, C. L. (1986). Action, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Grosz, B. J. and Sidner, C. L. (1990). Plans for discourse. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, SDF Benchmark Series, pages 417–444. MIT Press.

Grosz, B. J., Sparck Jones, K., and Webber, B. L., editors (1986). *Readings in Natural Language Processing.* Morgan Kaufmann Publishers.

Hayes, P. J. (1975). A representation for robot plans. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '75)*, pages 181–188. Reprinted in (Allen, Hendler and Tate, 1990).

Heeman, P. A. (1991). A computational model of collaboration on referring expressions. Master's Thesis, Technical Report CSRI 251, Department of Computer Science, University of Toronto.

Horrigan, M. K. (1977). Modelling simple dialogs. Master's Thesis, Technical Report 108, Department of Computer Science, University of Toronto.

Horton, D. and Hirst, G. (1991). Discrepancies in discourse models and miscommunication in conversation. In *Working Notes of the AAAI symposium: Discourse Structure in Natural Language Understanding and Generation*, pages 31–32.

Lambert, L. and Carberry, S. (1991). A tripartite plan-based model for dialogue. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 47–54.

Levelt, W. J. M. (1989). *Speaking: from intention to articulation.* Cambridge University Press, Cambridge.

Litman, D. J. and Allen, J. F. (1987). A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11(2):163–200.

Lochbaum, K. E., Grosz, B. J., and Sidner, C. L. (1990). Models of plans to support communication: An initial report. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '90)*, pages 485–490.

McRoy, S. W. and Hirst, G. (1991). An abductive account of repair in conversation. In *Working Notes of the AAAI symposium: Discourse Structure in Natural Language Understanding and Generation*, pages 52–57.

Mellish, C. S. (1985). *Computer Interpretation of Natural Language Descriptions.* Ellis Horwood Series in Artificial Intelligence. Ellis Horwood, Chichester, West Sussex, England.

Moore, J. D. and Swartout, W. R. (1991). A reactive approach to explanation: taking the user's feedback into account. In Paris, C. L., Swartout, W. R., and Mann, W. C., editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 3–48. Kluwer Academic Publishers.

Nadathur, G. and Joshi, A. K. (1983). Mutual beliefs in conversational systems: Their role in referring expressions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '83)*, pages 603-605.

Perrault, C. R. (1990). An application of default logic to speech act theory. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, SDF Benchmark Series, pages 161–185. MIT Press.

Perrault, C. R. and Cohen, P. R. (1981). It's for your own good: a note on inaccurate reference. In Joshi, A. K., Webber, B. L., and Sag, I., editors, *Elements of Discourse Understanding*, pages 217–230. Cambridge University Press, Cambridge.

Pollack, M. E. (1990). Plans as complex mental attitudes. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, SDF Benchmark Series, pages 77–103. MIT Press.

Reiter, E. (1990). The computational complexity of avoiding conversational implicature. In *Proceedings of the 28^{th} Annual Meeting of the Association for Computational Linguistics*, pages 97–104.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language.* Cambridge University Press, Cambridge.

Searle, J. R. (1990). Collective intentions and actions. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, SDF Benchmark Series, pages 401–415. MIT Press.

Svartvik, J. and Quirk, R. (1980). *A Corpus of English Conversation.* Lund Studies in English. 56. C.W.K. Gleerup, Lund.

Traum, D. R. (1991). Towards a computational theory of grounding in natural language conversation. Technical Report 401, Department of Computer Science, University of Rochester.

Walker, M. and Whittaker, S. (1990). Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28^{th} Annual Meeting of the Association for Computational Linguistics*, pages 70–78.

Webber, B. L. (1983). So what can we talk about now? In Brady, M. and Berwick, R. C., editors, *Computational Models of Discourse*, pages 331–371. MIT Press, Cambridge.

Wilensky, R. (1981). A model for planning in complex situations. *Cognition and Brain Theory*, 4. Reprinted in (Allen, Hendler and Tate, 1990).

Wilkens, D. E. (1985). Recovering from execution errors in SIPE. *Computational Intelligence*, 1:33–45. Reprinted in (Allen, Hendler and Tate, 1990).